IN THE CLAIMS:

Please amend the claims as set forth below.

1-31 (Cancelled)

32. (New) A processor comprising:

a plurality of decoders configured to decode instructions;

an alignment unit coupled to the plurality of decoders and to receive a plurality of instruction bytes fetched from a location identified by a fetch address and configured to align instructions within the plurality of instruction bytes to the plurality of decoders; and

a line predictor coupled to receive the fetch address and coupled to the alignment unit, the line predictor including a first memory comprising a plurality of entries, wherein each entry of the plurality of entries comprises a plurality of fields, and wherein each field of the plurality of fields is fixedly assigned to a respective decoder of the plurality of decoders by the physical position of the field within the entry and independent of the fetch address, and wherein the line predictor is configured to select a first entry of the plurality of entries, the first entry corresponding to the fetch address, and wherein each field of the plurality of fields in the first entry includes a respective instruction pointer of a first plurality of instruction pointers stored in the first entry, and wherein the respective instruction pointer, if valid, locates an instruction within the plurality of instruction bytes fetched from the location identified by the fetch address, and wherein the alignment unit is configured to align the instruction located by the respective instruction pointer to the respective decoder due to the respective instruction pointer being included in the field assigned to the respective decoder.

33. (New) The processor as recited in claim 32 wherein the first entry is further configured to store a next entry indication identifying a second entry of the plurality of entries within the first memory, wherein the line predictor is configured to subsequently select the second entry to provide a second plurality of instruction pointers stored therein responsive to the next entry indication.

34. (New) The processor as recited in claim 33 further comprising an instruction cache coupled to the line predictor, wherein the next entry indication further includes a next fetch address, and wherein the instruction cache is coupled to receive the next fetch address from the line predictor and to provide a second plurality of instruction bytes in response thereto.

35. (New) The processor as recited in claim 34 wherein the instruction cache is set associative, and wherein the first entry is further configured to store a way prediction corresponding to the next fetch address, and wherein the way prediction identifies which one of a plurality of ways within the instruction cache is to provide the second plurality of instruction bytes.

36. (New) The processor as recited in claim 35 wherein the instruction cache is configured to provide one or more of the second plurality of instruction bytes from a second storage location therein, and wherein the first entry includes a second way prediction corresponding to the second storage location.

37. (New) The processor as recited in claim 32 wherein the first entry is further configured to store control information corresponding to the instructions located by the first plurality of instruction pointers.

38. (New) The processor as recited in claim 37 wherein the control information includes an indication that at least one byte of a last instruction located by the first plurality of instruction pointers is stored on a different page than the plurality of instruction bytes.

39. (New) The processor as recited in claim 38 further comprising a translation lookaside buffer (TLB) configured to translate a second fetch address corresponding to the at least one byte.

40. (New) The processor as recited in claim 39 wherein the processor is configured to fetch the at least one byte from the different page.

41. (New) The processor as recited in claim 37 further comprising:

an instruction cache configured to store instruction bytes; and

a translation lookaside buffer (TLB) coupled to the instruction cache and configured to translate virtual addresses to physical addresses;

wherein the fetch address is a virtual address, and wherein the TLB is configured to translate the fetch address to a corresponding physical address and to provide the corresponding physical address to the instruction cache to fetch the plurality of instruction bytes from the location identified by the fetch address.

42. (New) The processor as recited in claim 41 wherein the virtual address comprises a linear address.

43. (New) The processor as recited in claim 33 wherein the line predictor further includes a second memory coupled to receive the fetch address and further coupled to the first memory, the second memory comprising a second plurality of entries configured to store fetch addresses and indexes into the first memory.

44. (New) The processor as recited in claim 43 wherein the second memory is configured to compare the fetch address to fetch addresses stored in the second plurality of entries and to select a second entry of the second plurality of entries in response to the

fetch address matching the fetch address stored in the second entry, and wherein the second memory is configured to provide the index stored in the second entry to the first memory to select the first entry.

45. (New) The processor as recited in claim 44 wherein the line predictor is configured to inhibit access to the second memory if the next entry indication in the first entry is valid.

46. (New) The processor as recited in claim 43 wherein the second memory comprises a content addressable memory (CAM).

47. (New) The processor as recited in claim 46 wherein the first memory comprises a random access memory (RAM).

48. (New) The processor as recited in claim 46 wherein the CAM is configured to compare a portion of the fetch address to the fetch addresses stored in the second plurality of entries.

49. (New) A method comprising:

generating a fetch address; and

selecting a first entry of a plurality of entries in a line predictor responsive to the fetch address, wherein each entry of the plurality of entries comprises a plurality of fields, and wherein each field of the plurality of fields is fixedly assigned to a respective decoder of a plurality of decoders by the physical position of the field within the entry and independent of the fetch address, and wherein each field of the plurality of fields in the first entry includes a respective instruction pointer of a first plurality of instruction pointers stored in the first entry, and wherein the respective instruction pointer, if valid, locates an instruction within a plurality of instruction

bytes fetched from a location identified by the fetch address, and wherein the instruction located by the respective instruction pointer is to be aligned to the respective decoder due to the respective instruction pointer being included in the field assigned to the respective decoder; and

aligning instructions from the plurality of instruction bytes to the plurality of decoders responsive to the first plurality of instruction pointers in a first plurality of fields of the first entry.

50. (New) The method as recited in claim 49 wherein the first entry is further configured to store a next entry indication, the method further comprising selecting a second entry of the plurality of entries responsive to the next entry indication.

51. (New) The method as recited in claim 50 wherein the next entry indication includes a next fetch address, the method further comprising:

providing the next fetch address to an instruction cache; and

accessing a first storage location in the instruction cache in response to the next fetch address.

52. (New) The method as recited in claim 51 wherein the instruction cache is set associative, and wherein the first entry is further configured to store a way prediction, the method further comprising selecting one of a plurality of ways of the instruction cache from which to fetch the plurality of instruction bytes in response to the way prediction.

53. (New) The method as recited in claim 52 wherein the first entry is further configured to store a second way prediction, the method further comprising:

accessing a second storage location in the instruction cache in response to the next fetch address; and

selecting one of the plurality of ways in response to the second way prediction.

54. (New) The method as recited in claim 49 wherein the first entry is further configured to store an indication that a last instruction located by the first plurality of instruction pointers includes at least one byte in a different page, the method further comprising:

generating a second fetch address corresponding to the different page;

translating the second fetch address; and

fetching instruction bytes from the instruction cache using the second fetch address.

55. (New) The method as recited in claim 49 wherein the line predictor further comprises a second memory including a second plurality of entries, each of the second plurality of entries storing a particular fetch address and a corresponding index into the first memory, wherein the selecting comprises:

comparing the fetch address to the particular fetch address stored in each of the second plurality of entries;

selecting the corresponding index from a second entry of the second plurality of entries in response to the comparing; and

selecting the first entry responsive to the corresponding index.

56. (New) The method as recited in claim 55 wherein the comparing comprises comparing a portion of the fetch address to a corresponding portion of the particular fetch address.

57. (New) The method as recited in claim 48 further comprising fetching the plurality of instructions using a physical address translated from the fetch address, the fetch address being a virtual address.

58. (New) A computer system comprising:

a processor comprising:

a plurality of decoders configured to decode instructions; and

an alignment unit coupled to the plurality of decoders and to receive a plurality of instruction bytes fetched from a location identified by the fetch address and configured to align instructions within the plurality of instruction bytes to the plurality of decoders; and

a line predictor coupled to receive the fetch address and coupled to the alignment unit, the line predictor including a first memory comprising a plurality of entries, wherein each entry of the plurality of entries comprises a plurality of fields, and wherein each field of the plurality of fields is fixedly assigned to a respective decoder of the plurality of decoders by the physical position of the field within the entry and independent of the fetch address, and wherein the line predictor is configured to select a first entry of the plurality of entries, the first entry corresponding to the fetch address, and wherein each field of the plurality of fields in the first entry includes a respective instruction pointer of a first plurality of instruction pointers stored in the first entry, and wherein the respective instruction pointer, if valid, locates an instruction within the plurality of instruction bytes fetched from the location identified by the fetch address, and wherein the alignment unit is configured to align the instruction located by the

respective instruction pointer to the respective decoder due to the respective instruction pointer being included in the field assigned to the respective decoder; and

an input/output (I/O) device configured to communicate between the computer system and another computer system to which the I/O device is couplable.

59. (New) The computer system as recited in claim 58 wherein the I/O device comprises a modem.

60. (New) A line predictor coupled to receive a fetch address, the line predictor including a first memory comprising a plurality of entries, wherein each entry of the plurality of entries comprises a plurality of fields, and wherein each field of the plurality of fields is fixedly assigned to a respective decoder of a plurality of decoders by the physical position of the field within the entry and independent of the fetch address, and wherein the line predictor is configured to select a first entry of the plurality of entries, the first entry corresponding to the fetch address, and wherein each field of the plurality of fields in the first entry includes a respective instruction pointer of a first plurality of instruction pointers stored in the first entry, and wherein the respective instruction pointer, if valid, locates an instruction within a plurality of instruction bytes fetched from a location identified by the fetch address, and wherein the instruction located by the respective instruction pointer is to be aligned to the respective decoder due to the respective instruction pointer being included in the field assigned to the respective decoder, and wherein the line predictor is configured to supply the first plurality of instruction pointers from the plurality of fields in the first entry to an alignment unit that is configured to align instructions from the plurality of instruction bytes to the plurality of decoders responsive to the first plurality of instruction pointers in the plurality of fields of the first entry.

61. (New) The line predictor as recited in claim 60 wherein the first entry is further configured to store a next entry indication identifying a second entry of the plurality of

entries within the first memory, wherein the line predictor is configured to subsequently select the second entry to provide a second plurality of instruction pointers stored therein responsive to the next entry indication.

62. (New) The line predictor as recited in claim 61 wherein the next entry indication further includes a next fetch address.

63. (New) The line predictor as recited in claim 62 wherein the first entry is further configured to store a way prediction corresponding to the next fetch address, and wherein the way prediction identifies which one of a plurality of ways within an instruction cache is to provide the second plurality of instruction bytes.

64. (New) The line predictor as recited in claim 63 wherein the first entry includes a second way prediction corresponding to a second storage location in the instruction cache.

65. (New) The line predictor as recited in claim 61 further comprising a second memory coupled to receive the fetch address and further coupled to the first memory, the second memory comprising a second plurality of entries configured to store fetch addresses and indexes into the first memory.

66. (New) The line predictor as recited in claim 65 wherein the second memory is configured to compare the fetch address to fetch addresses stored in the second plurality of entries and to select a second entry of the second plurality of entries in response to the fetch address matching the fetch address stored in the second entry, and wherein the second memory is configured to provide the index stored in the second entry to the first memory to select the first entry.

67. (New) The line predictor as recited in claim 66 further configured to inhibit access to the second memory if the next entry indication in the first entry is valid.

68. (New) The line predictor as recited in claim 60 wherein the first entry includes an indication that at least one byte of a last instruction located by the first plurality of instruction pointers is stored on a different page than the plurality of instruction bytes.